Name:

ID:

CSCI 3104, Algorithms **Profs. Chen & Grochow**

Problem Set 7 – Due Thur Mar 12 11:55pm **Spring 2020, CU-Boulder**

Advice 1: For every problem in this class, you must justify your answer: show how you arrived at it and why it is correct. If there are assumptions you need to make along the way, state those clearly.

Advice 2: Informal reasoning is typically insufficient for full credit. Instead, write a logical argument, in the style of a mathematical proof.

Instructions for submitting your solutions:

- All submissions must be typed.
- You should submit your work through the **class Canvas page** only.
- You may not need a full page for your solutions; pagebreaks are there to help Gradescope automatically find where each problem is. Even if you do not attempt every problem, please allot at least as many pages per problem (or subproblem) as are allotted in this template.

Quicklinks: 1 2 3 3a 3b 3c 3d

Name:

ID:

CSCI 3104, Algorithms **Profs. Chen & Grochow**

Problem Set 7 – Due Thur Mar 12 11:55pm **Spring 2020, CU-Boulder**

1. *Let $G(V, E, w)$ be a weighted graph. The* **Cycle Property** *provides conditions for when an edge is* ***not*** *included in any minimum spanning tree (MST) of G. The* ***Cycle Property*** *is stated as follows. Let C be a cycle in G, and let edge $e = (u, v)$ be a maximum-cost edge on C. Then the edge e does not belong to any MST of G.*

   *Use an exchange argument to show why this property holds. You may assume that all edge costs are distinct. [**Note:** You may freely use any of the tree properties covered in Michael's Tree Notes on Canvas.]*

   Proof: If G has no cycles, then the Cycle Property holds true vacuously. So suppose G has at least one cycle C. Let $e \in C$ be the heaviest edge on C, and let T be a spanning tree of G containing e. As T is a tree and $e \in T$, e is a cut edge of T. So $T - e$ contains two connected components S and T. As C is a cycle, there exists an edge $e' \in C$ such that $e' \notin T$ (otherwise, T would contain the cycle C, contradicting the fact that trees have no cycles). We note that as e' and e lie on the same cycle C, e' connects S and T. So $(T - e) \cup e'$ is again a spanning tree. As the edge weights of G are distinct and e is the heaviest edge on C, weight(e') < weight(e). So weight$((T - e) \cup e')$ < weight(T). The result follows.

2. *Let $G(V, E, w)$ be a connected, weighted graph. A cut of G is a set $C \subset E$ of edges such that removing the edges of C from G disconnects the graph. The* **Cut Property** *provides conditions for when an edge* ***must*** *be included in a minimum spanning tree (MST) of G. The* ***Cut Property*** *is stated as follows. Let $C \subset E$ be a cut. Suppose $e = (u, v)$ is the minimum-weight edge in C. Then every MST contains the edge e.*

   *Use an exchange argument to show why this property holds. You may assume that all edge costs are distinct. [**Note:** You may freely use any of the tree properties covered in Michael's Tree Notes on Canvas.]*

   Proof: Let C be minimal a cut of G. We note that as C is a cut, $G - C$ is not connected (this is the definition of a cut). So we partition the vertices of G into sets S and $V - S$, such that if $i \in S$ and $j \in V - S$ are adjacent vertices in G, then $(i, j) \in C$. It follows that any spanning tree of G requires at least one edge of C. Suppose $u, x \in V$ belong to the same connected component of $G - C$, $v, y \in V - S$ belong to the same connected component of $G - C$, and $(x, y) \in E$. So $(x, y) \in C$. We note that we may include either (x, y) or $e = (u, v)$, but not both edges in our spanning tree, as including both edges would create a cycle. As edge weights are distinct and e is the minimum weight edge of C, including $e = (u, v)$ results in a lower weight spanning tree than including (x, y). The result follows.

   Remark: We include the diagram on the next page to help illustrate the concept. Here, the green circles represent the connected components of S, and the blue circles represent the connected components of $V - S$. The black dotted is used to help illustrate that the edges of C cross this line.

   Including the red (u, v) edge and the purple (x, y) edge creates a cycle. So we may only include (x, y) or (u, v). We opt to include (u, v), as (u, v) has smaller weight than (x, y).